What Information Theory says about Best Response and about Binding Contracts

David H. Wolpert
NASA Ames Research Center,
Moffett Field, CA, 94035, USA
dhw@email.arc.nasa.gov

Product Distribution (PD) theory is the information-theoretic extension of conventional full-rationality game theory to bounded rational games. Here PD theory is used to investigate games in which the players use bounded rational best-response strategies. This investigation illuminates how to determine the optimal organization chart for a corporation, or more generally how to order the sequence of moves of the players / employees so as to optimize an overall objective function. It is then shown that in the continuum-time limit, bounded rational best response games result in a variant of the replicator dynamics of evolutionary game theory. This variant is then investigated for team games, in which the players share the same utility function, by showing that such continuum-limit bounded rational best response is identical to Newton-Raphson iterative optimization of the shared utility function. Next PD theory is used to investigate changing the coordinate system of the game, i.e., changing the mapping from the joint move of the players to the arguments in the utility functions. Such a change couples those arguments, essentially by making each players' move be an offered binding contract.

I. INTRODUCTION

Recent work has shown that information theory [1–3] provides a principled extension of noncooperate conventional game theory to accommodate bounded rationality [4]. Intuitively, this extension is based on Occam's razor: Given only partial knowledge concerning a game's (bounded rational) equilibrium, introduce as little extra information as possible beyond that partial knowledge in inferring the joint mixed strategy of that equilibrium. This is formalized by setting the joint mixed strategy of the game's equilibrium, $q(x \in X) = \prod_i q_i(x_i)$, to the minimizer of a set of Lagrangian functions.

The field of Probability Collectives concerns the optimization of distributions over the variables of interest, rather than the optimization of those variables directly. The special case considered here, where the joint distribution over the variables of interest is a product distribution, is known as Product Distribution (PD) theory [5–11]. This paper uses PD theory to investigate several aspects of game theory not considered in [4]. PD theory is applied to games in which the players use bounded rational versions of best-response strategies. It is also used to investigate changing the coordinate system of a game, i.e., changing the mapping from the joint move of the players to the arguments in the utility functions. Such changes couple those arguments, essentially by making the players' moves be offered binding contracts. This paper also uses PD theory to illuminate recent work in adaptive distributed control.

Sec. II reviews how information theory can be used to derive bounded rational noncooperative game theory. Some simple examples of bounded rational games are then presented.

Sec. III analyzes scenarios in which the players use bounded rational versions of best response strategies. Particular attention is paid to team games, in which the players share the same utility function. The analysis for this case provide insight into how to optimize the sequence of moves by the players, as far as their shared utility is concerned. This can be viewed as a formal way to optimize the organization chart of a corporation.

Best response strategies, even bounded rational ones, are poor models of real-world computational players that use Reinforcement Learning (RL) [12–15]. Sec. IV considers iterated games in which players use a (bounded rational) variant of best response, a variant that is more realistic for computational players, and arguably for human players as well. In this variant the conditional expected utilities used by each player to update her strategy is a decaying average of recent conditional expected utilities; this implements a bias by the player to dampen large and sudden changes in her strategy. This variant is then explored for the case of team games. The continuum limit of the dynamics of such games is shown to be variant of the replicator dynamics. It is shown such continuum-limit bounded rational best response is identical to Newton-Raphson iterative optimization of the shared utility function of such games.

The next section investigates changing the coordinate system of the game. By doing this the moves of the players get transformed, into (bounded rational) contracts binding them. Some of the implications for optimal organization charts of such bounded rational contracts are elucidated, as well as their use to speed convergence in team games.

This paper ends with a discussion of how these results relate to other work, and with a brief overview of extensions of these resuls.

II. PD THEORY AS BOUNDED RATIONAL NONCOOPERATIVE GAME THEORY

In this section we motivate PD theory as the information-theoretic formulation of bounded rational game theory. We use the integral sign ($\int$) with the associated measure implicit, i.e., it indicates sums if appropriate, Lebesgue integrals over $\mathbb{R}^n$ if appropriate, etc. In addition, the subscript (i) is used to indicate all index values other than i. Finally, we use $\mathcal{P}$ to indicate the set of all probability distributions over a vector space, and $\mathcal{Q}$ to indicate the subset of $\mathcal{P}$ consisting of all product distributions (i.e., the associated Cartesian product of unit simplices).

A. Review of noncooperative game theory

In noncooperative game theory one has a set of N **players**. Each player i has its own set of allowed **pure strategies**. A **mixed strategy** is a distribution $q_i(x_i)$ over player i's possible pure strategies. Each player i also has a **private utility** function g_i that maps the pure strategies adopted by all N of the players into the real numbers. So given mixed strategies of all the players, the expected utility of player i is $E(g_i) = \int dx \, \prod_j q_j(x_j) g_i(x)$ [44].

In a **Nash equilibrium** every player adopts the mixed strategy that maximizes its expected utility, given the mixed strategies of the other players. More formally, $\forall i, q_i = \text{argmax}_{q'_i} \int dx \; q'_i \prod_{j \neq i} q_j(x_j) \; g_i(x)$. Perhaps the major objection that has been raised to the Nash equilibrium concept is its assumption of **full rationality** [16–20]. This is the assumption that every player i can both calculate what the strategies $q_{j \neq i}$ will be and then calculate its associated optimal distribution. In other words, it is the assumption that every player will calculate the entire joint distribution $q(x) = \prod_j q_j(x_j)$.

In the real world, this assumption of full rationality almost never holds, whether the players are humans, animals, or computational agents [11, 16, 21–27]. This is due to the cost of computation of that optimal distribution, if nothing else. This real-world **bounded rationality** is one of the major impediments to applying conventional game theory in the real world.

B. Review of the minimum information principle

Shannon was the first person to realize that based on any of several separate sets of very simple desiderata, there is a unique real-valued quantification of the amount of syntactic information in a distribution $P(y)$. He showed that this amount of information is the negative of the Shannon entropy of that distribution, $S(P) = -\int dy \;\; P(y)\ln[\frac{P(y)}{\mu(y)}]$. So for example, the distribution with minimal information is the one that doesn't distinguish at all between the various y, i.e., the uniform distribution. Conversely, the most informative distribution is the one that specifies a single possible y. Note that for a product distribution, entropy is additive, i.e., $S(\prod_i q_i(y_i)) = \sum_i S(q_i)$.

Say we given some incomplete prior knowledge about a distribution $P(y)$. How should one estimate $P(y)$ based on that prior knowledge? Shannon's result tells us how to do that in the most conservative way: have your estimate of $P(y)$ contain the minimal amount of extra information beyond that already contained in the prior knowledge about $P(y)$. Intuitively, this can be viewed as a version of Occam's razor: introduce as little extra information beyond that you are provided in your inferring of P. This minimum information approach is called the maxent principle. It has proven extremely powerful in domains ranging from signal processing to supervised learning [2]. In particular, it is has been successfully used in many statistics applications, including econometrics [28]. It has even provided what many consider the cleanest derivation of the foundations of statistical physics [29].

C. Maxent Lagrangians

Much of the work on equilibrium concepts in game theory adopts the perspective of an external observer of a game. We are told something concerning the game, e.g., its cost functions, information sets, etc., and from that wish to predict what joint strategy will be followed by real-world players of the game. Say that in addition to such information, we are told the expected utilities of the players. What is our best estimate of the distribution q that generated those expected cost values? By the maxent principle, it is the distribution with maximal entropy, subject to those expectation values.

To formalize this, for simplicity assume a finite number of players and of possible strategies for each player. To agree with the convention in fields other than game theory (e.g., optimization, statistical physics, etc.), from now on we implicitly flip the sign of each g_i so that the associated player i wants to minimize that function rather than maximize it. Intuitively, this flipped $g_i(x)$ is the "cost" to player i when the joint-strategy is x.

With this convention, given prior knowledge that the expected utilities of the players are given by the set of values $\{\epsilon_i\}$, the maxent estimate of the associated q is given by the minimizer of the Lagrangian

$$\mathscr{L}(q) \;\equiv\; \sum_i \beta_i[E_q(g_i) - \epsilon_i] - S(q) \tag{1}$$

$$\;=\; \sum_i \beta_i[\int dx \; \prod_j q_j(x_j) g_i(x) - \epsilon_i] - S(q) \tag{2}$$

where the subscript on the expectation value indicates that it evaluated under distribution q. The $\{\beta_i\}$ are "inverse temperatures" implicitly set by the constraints on the expected utilities.

Solving, we find that the mixed strategies minimizing the Lagrangian are related to each other via

$$q_i(x_i) \propto e^{-E_{q_{(i)}}(G|x_i)} \tag{3}$$

where the overall proportionality constant for each i is set by normalization, and $G \equiv \sum_i \beta_i g_i$ [45]. In Eq. 3 the probability of player i choosing pure strategy x_i depends on the effect of that choice on the utilities of the other players. This reflects the fact that our prior knowledge concerns all the players equally.

If we wish to focus only on the behavior of player i, it is appropriate to modify our prior knowledge. To see how to do this, first consider the case of maximal prior knowledge, in which we know the actual joint-strategy of the players, and therefore all of their expected costs. For this case, trivially, the maxent principle says we should "estimate" q as that joint-strategy (it being the q with maximal entropy that is consistent with our prior knowledge). The same conclusion holds if our prior knowledge also includes the expected cost of player i.

Modify this maximal set of prior knowledge by removing from it specification of player i's strategy. So our prior knowledge is the mixed strategies of all players other than i, together with player i's expected cost. We can incorporate prior knowledge of the other players' mixed strategies directly, without introducing Lagrange parameters. The resultant **maxent Lagrangian** is

$$\begin{aligned}\mathscr{L}_i(q_i) &\equiv \beta_i[\epsilon_i - E(g_i)] - S_i(q_i) \\ &= \beta_i[\epsilon_i - \int dx \prod_j q_j(x_j) g_i(x)] - S_i(q_i)\end{aligned}$$

solved by a set of coupled **Boltzmann distributions**:

$$q_i(x_i) \propto e^{-\beta_i E_{q_{(i)}}(g_i|x_i)}. \qquad (4)$$

Following Nash, we can use Brouwer's fixed point theorem to establish that for any non-negative values $\{\beta\}$, there must exist at least one product distribution given by the product of these Boltzmann distributions (one term in the product for each i).

The first term in $\mathscr{L}_i$ is minimized by a perfectly rational player. The second term is minimized by a perfectly *irrational* player, i.e., by a perfectly uniform mixed strategy q_i. So β_i in the maxent Lagrangian explicitly specifies the balance between the rational and irrational behavior of the player. In particular, for $\beta \to \infty$, by minimizing the Lagrangians we recover the Nash equilibria of the game. More formally, in that limit the set of q that simultaneously minimize the Lagrangians is the set of mixed strategy equilibria of the game, together with the set of delta functions about the pure Nash equilibria of the game. The same is true for Eq. 3.

Note also that independent of information-theoretic considerations, the Boltzmann distribution is a reasonable (highly abstracted) model of how human players will behave. Typically humans do some "exploration" as well as "exploitation", trying out all moves, with frequency as the expected cost of the move increases. This is captured in the Boltzmann distribution mixed strategy.

One can formalize the concept of the rationality of a player in a way that applies to any distribution, not just a Boltzmann distribution. One does this with a **rationality operator** which maps a q and a g_i to a non-negative real value measuring the rationality of player i in adopting strategy q_i given private cost function g_i and strategies $q_{(i)}$ of the other players. For the solution in Eq. 4 and private cost g_i, the value of that operator is just β_i [4].

Eq. 3 is just a special case of Eq. 4, where all player's share the same private cost function, G. (Such games are known as **team games**.) This relationship reflects the fact that for this case, the difference between the maxent Lagrangian and the one in Eq. 2 is independent of q_i. Due to this relationship, our guarantee of the existence of a solution to the set of maxent Lagrangians implies the existence of a solution of the form Eq. 3. Typically players will be closer to minimizing their expected cost than maximizing it. For prior knowledge consistent with such a case, the β_i are all non-negative.

For each player i define

$$f_i(x, q_i(x_i)) \equiv \beta_i g_i(x) + \ln[q_i(x_i)].$$

Then we can write the maxent Lagrangian for player i as

$$\mathscr{L}_i(q) = \int dx\ q(x) f_i(x, q_i(x_i)). \qquad (6)$$

Now in a bounded rational game every player sets its strategy to minimize its Lagrangian, given the strategies of the other players. In light of Eq. 6, this means that we can interpret each player in a bounded rational game as being perfectly rational for a cost function that incorporates its computational cost. To do so we simply need to expand the domain of "cost functions" to include (logarithms of) probability values as well as joint moves.

D. Examples of bounded rational equilibria

It can be difficult to start with a set of cost functions and associated rationalities β_i and then solve for the associated bounded rational equilibrium q. Solving for q when prior knowledge consists of expected costs ϵ_i rather than rationalities can be even more tedious. (In that situation the β_i are not specified upfront but instead are Lagrange parameters that we must solve for.) However there is an alternative approach to constructing examples of games and their bounded rational equilibria that is quite simple. In this alternative one starts with a particular mixed strategy q and then solves for a game for which q is a bounded rational equilibrium, rather than the other way around.

To illustrate this, consider a 2-person noncooperative single-stage game. Let each player have 3 possible moves. Indicate each players' three possible moves by the numerals 0, 1, and 2. Say the (bounded rational) mixed strategy equilibrium is

$$\begin{aligned}&q_1(0) = 1/2,\ q_1(1) = 1/4,\ q_1(2) = 1/4; \\ &q_2(0) = 2/3,\ q_2(1) = 1/4,\ q_2(2) = 1/12\ . \qquad (7)\end{aligned}$$

Now we know that at the equilibrium, $q_1(x_1) \propto e^{-\beta_1 E(g_1|x_1)}$, where β_1 is player 1's rationality, and g_1 is her cost function (the negative of her cost function). This means for example that

$$e^{-(\beta_1[E(g_1|x_1=0)-E(g_1|x_1=1)])} = \frac{q_1(0)}{q_1(1)} \;=\; 2,$$

i.e.,

$$\beta_1[E(g_1 \mid x_1 = 0) - E(g_1 \mid x_1 = 1)] \;=\; -\ln(2). \quad (8)$$

We have a similar equation for the remaining independent difference in expectation values for player 1. The analogous pair of equations for player 2 also hold.

Now define the vectors $\mathbf{g}_{i;j}(.) \equiv g_i(x_i = j, .)$. So for example $\mathbf{g}_{1;0} = (g_1(x_1 = 0, x_2 = 0), g_1(x_1 = 0, x_2 = 1), g_1(x_1 = 0, x_2 = 2))$. Then we can express our equations compactly as four dot product equalities:

$$\begin{aligned}
\beta_1(\mathbf{g}_{1;0} - \mathbf{g}_{1;1}) \cdot q_2 &= -\ln(2), \\
\beta_1(\mathbf{g}_{1;0} - \mathbf{g}_{1;2}) \cdot q_2 &= -\ln(2) \;; \\
\beta_2(\mathbf{g}_{2;0} - \mathbf{g}_{2;1}) \cdot q_1 &= -\ln(8/3), \\
\beta_2(\mathbf{g}_{2;0} - \mathbf{g}_{2;2}) \cdot q_1 &= -\ln(8) \;. \qquad (9)
\end{aligned}$$

Note that we can absorb each β_i into its associated g_i; all that matters is their product.

We can now plug in for the vectors q_1 and q_2 from Eq. 7 and simply write down a set of solutions for the four three-dimensional vectors $\mathbf{g}_{i,j}$. For these $\{g_i\}$ the bounded ratinal equilibrium is given by the q of Eq. 7. If desired, we can evaluate the associated expected values of the cost functions for the two players; our q is the bounded ratinal equilibrium for thos eexpected costs.

Note that the variables in the first pair of equalities in Eq. 9 are independent of those in the second pair. In other words, whereas the Boltzmann equations giving q for a specified set of g_i are a set of coupled equations, the equations giving the g_i for a specified q are not coupled. Note also that our equations for the $\mathbf{g}_{i;j}$ are (extremely) underconstrained. This illustrates how compressive the mapping from the g_i to the associated equilibrium q is. Bear in mind though that that mapping is also multivalued in general; in general a single set of cost functions can have more than one equilibrium, just like it can have more than one Nash equilibrium.

The generalization of this example to arbitrary numbers of players with arbitrary move spaces is immediate. As before, indicate the moves of every player by an associated set of integer numerals starting at 0. Recall that the subscript (i) on a vector indicate all components but the i'th one. Also absorb the rationalities β_i into the associated g_i.

Now specify q and the vectors $g_i(x_i = 0, .)$ (one vector for each i) to be anything whatsoever. Then for all players i, the only associated constraint on the i'th cost function concerns certain projections of the vectors $g_i(x_i > 0, .)$ (one projection for each value $x_i > 0$). Concretely, $\forall i, x_i > 0$,

$$\int dx'_{(i)}\; g_i(x_i, x'_{(i)}) \prod_{j \neq i} q_j(x'_j) \;=\; -\ln(\frac{q_i(0)}{q_i(x_i)}) + \int dx'_{(i)} g_i(0, x'_{(i)}) \prod_{j \neq i} q_j(x'_j), \quad (10)$$

i.e., $\forall i, x_i > 0$,

$$\mathbf{g}_i(x_i, .) \cdot q_{(i)} \;=\; -\ln(\frac{q_i(0)}{q_i(x_i)}) + \mathbf{g}_i(0, .) \cdot q_{(i)}. \quad (11)$$

All the terms on the right-hand side are specified, as well as the $q_{(i)}$ term on the left-hand side. Any $\mathbf{g}_i(x_i, .)$ that obeys the associated equation has the specified q as a bounded rational equilibrium.

E. Discussion

There are numerous alternative interpretations of the information-theoretic formulation of bounded rationality presented here. For example, change our prior knowledge to be the entropy of each player i's strategy, i.e., how unsure it is of what move to make. Now we cannot use information theory to make our estimate of q. Given that players try to minimize expected cost, a reasonable alternative is to predict that each player i's expected cost will be as small as possible, subject to that provided value of the entropy and the other players' strategies. The associated Lagrangians are $\alpha_i[S(q_i) - \sigma_i] - E(g_i)$, where σ_i is the provided entropy value. This is equivalent to the maxent Lagrangian, and in particular has the same solution, Eq. 4.

Another alternative interpretation involves **world cost** functions, which are quantifications of the quality of a joint pure strategy x from the point of view of an external observer (e.g., a system designer, the government, an auctioneer, etc.). A particular class of world cost functions are (negatives of) "social welfare functions", which can be expressed in terms of the cost functions of the individual players. Perhaps the simplest example is $G(x) = \sum_i \beta_i g_i(x)$, where the β_i serve to trade off how much we value one player's cost vs. anothers. If we know the value of this social welfare function, but nothing else, then maxent tells us to minimize the Lagrangian of Eq. 2.

Often our prior knowledge will not consist of exact specification of the expected costs of the players, even if that knowledge arises from watching the players make their moves. Such alternative kinds of prior knowledge are addressed in [4, 6]. In particular, in those references it is shown how one might define a "rationality operator" that quantifies the rationality of any pair of a player's mixed strategy and cost function, given the mixed strategey of all the other players. If one's prior knowledge is the the values of the rationalities of the players, then one again arises at solutions of the form in

Eq. 4, where the value of β_i reflects the rationality of that player.

In addition, in the real world the information we are provided concerning the system often will not consist of *exact* values of functionals of q, be those values expected costs, rationalities, or what have you. Rather that knowledge will be in the form of data, D, together with an associated likelihood function over the space of q. For example, that knowledge might consist of a bias toward particular rationality values, rather than precisely specified values:

$$P(D \mid q) \propto e^{-\alpha \sum_i [R_{KL}([g_i]_{i,q}) - \rho_i]^2}.$$

where α sets the strength of the bias.

As mentioned in the introduction, these results can also be extended in many ways (e.g., to allow multiple cost functions, variables numbers of players, etc.). Some such extensions are explored below.

III. BOUNDED RATIONAL VERSIONS OF BEST RESPONSE

One crude way to try to find the q given by Eq. 4 would be an iterative process akin to the best-response scheme of game theory [16]. Given any current distribution q, in this scheme all agents i simultaneously replace their current distributions. In this replacement each agent i replaces q_i with the distribution given in Eq. 4 *based on the current* $q_{(i)}$. This scheme is the basis of the use of Brouwer's fixed point theorem to prove that a solution to Eq. 4 exists. Accordingly, it is called **parallel Brouwer updating**. (This scheme goes by many names in the literature, from Boltzmann learning in the RL community to block relaxation in the optimization community.)

Sometimes the conditional expected cost for each agent can be calculated explicitly at each iteration. More generally, it must be estimated. This can be done via Monte-Carlo sampling, iterated across a block of time throughout which q is unchanging. During that block the agents all repeatedly and jointly IID sample their probability distributions to generate joint moves, and the associated cost values are recorded. These are then use to estimate all the conditional expected costs, which are then used to determine the parallel Brouwer update[46].

This is exactly what is done in RL-based schemes in which each agent maintains a data-based estimate of its cost for each of its possible moves, and then chooses its actual move stochastically, by sampling a Boltzmann distribution of those estimates. (See [5] for ways to get accurate MC estimates more efficiently than in this simple scheme, e.g., by exploiting the bias-variance tradeoff of statistics.)

One alternative to parallel Brouwer updating is **serial** Brouwer updating, where we only update one q_i at a time. This is analogous to a Stackelberg game, in that one agent makes its move and then the other(s) respond [17, 19]. In a team game, any serial Brouwer updating must reduce the common Lagrangian, in contrast to the case with parallel Brouwer updating.

There are many versions of serial updating. In **cyclic** serial Brouwer updating, one cycles through the i in order. In **random** serial Brouwer updating, one cycles through them in a random fashion.

In **greedy** serial Brouwer updating, instead of cycling through all i, at each iteration we choose what player to update based on how much that will reduce the common Lagrangian. Those reductions can be evaluated without explicitly calculating the associated Boltzmann distributions. To see how, use N_i to indicate the normalization constant of Eq. 4. Then define the **Lagrangian gap** at q for player i as $\ln[N_i] + \int dx_i q_i(x_i) E_{q_{(i)}}(g_i \mid x_i) + \int dx_i q_i(x_i) \ln[q_i(x_i)]$. This is how much $\mathscr{L}$ is reduced if only q_i undergoes the Brouwer update [47].

Another obvious variant of these schemes is mixed serial/parallel Brouwer updating, in which one subset of the players moves in synchrony, followed by another subset, and so on. Such updating in a team game can be viewed as a simple model of the organization chart of the players. For example, this is the case when the players are a corporation, with G being a common cost function based on the corporation's performance.

Say we observe the functioning of such an organization over time, and view those observations as Monte Carlo sampling of its behavior. Then we use those samples to statistically estimate how best to do serial/parallel Brouwer updating, for the purpose of minimizing the shared cost function G. This can be viewed as a way to optimize the organization chart coupling the players.

IV. PARALLEL BROUWER WITH DATA-AGING IS NEAREST NEWTON

This section considers a variant of best-response that is more realistic (more accurately modeling RL-based computational players that are actually used in machine learning, and arguably more accurately modeling human players as well). In this variant the expected cost used by each player to update her strategy is a decaying average of recent expected utilities; this decay reflects a conservative preference for dampening large changes in strategy.

Such a bias is used (implicitly or otherwise) in most multi-player RL algorithms. For example, in the COIN framework each agent i collects a data set of pairs of what value its private cost function has at timestep t together with the move it made then. It then estimates its cost for move x_i as a weighted average of all the cost values in its data set for that move. The weights are exponentially decaying functions of how long ago the associated observation was made. This **data-aging** is crucial to reflect the non-stationarity of agent i's environment, i.e., the fact that the other agents are changing their strategies with time. Arguably similar modifications to best response are used by human players. Indeed, in idealized learning rules like ficticious play, such dampening is crucial.

A. The dynamics of Brouwer updating

Consider a multi-stage game where at the end of iteration t, each player i updates her distribution $q_i(.,t)$ to

$$q_i(x_i,t) \;=\; \frac{e^{-\Phi_i(x_i,t)}}{\int dx'_i e^{-\Phi_i(x'_i,t)}}. \qquad (12)$$

This is a generalization of parallel Brouwer updating, where the function being exponentiated can be Q values (as in Q-learning [30]), single-instant reward values, distorted versions of these (e.g., to incorporate data-aging), etc.

As an example, for single-instant rewards (i.e., conventional parallel Brouwer), $\Phi_i(x_i,t)$ is player i's estimate of (β_i times) her conditional expected cost for taking move x_i at time $t-1$. If that estimate were exact, this would mean

$$\begin{aligned}\Phi_i(x_i,t) \;&=\; \beta E(g_i \mid x_i) \\ &=\; \beta \int dx_{(i)} q_{(i)}(x_{(i)}, t-1) g_i(x_i, x_{(i)}). \qquad (13)\end{aligned}$$

As another example, for Q-learning, one player is Nature and her distribution is always a delta function. In this case $\Phi_i(x_i,t)$ is the Q-value for player i taking action x_i, when the state of Nature is as specified by the associated delta function in $q(.,t-1)$.

Note that there's no Monte Carlo sampling being done here, as there is in most real-world RL; this is a somewhat abstracted version of such RL. Alternatively, the analysis here becomes exact when Φ_i is evaluated closed form, or (as when Φ_i is an empirical expectation value) there's enough samples in a Monte Carlo block so that empirical averages effectively give us exact values of expected quantities.

At this point we have to say something about how Φ_i evolves with time. Consider the case where Φ_i is an estimate of some function ϕ_i, formed by exponential aging of the previous ϕ values. In our case (since everything is evaluated closed form) assuming there have been an infinite number of preceding timesteps, this is the same as geometric data-aging:

$$\Phi_i(x_i,t) = \alpha\phi_i(x_i, q(t-1)) + (1-\alpha)\Phi_i(x_i, t-1) \quad (14)$$

for some appropriate function ϕ_i [48]. For example, in parallel Brouwer updating, $\phi_i(x_i,t) = \beta E(g_i \mid x_i, q_{(i)}(t))$, while $\Phi_i(x_i,t)$ is a geometric average of the previous values of $\phi(x_i)$.

B. The continuum-time limit

To go to the continuum-time limit, let t be a real variable, and replace the temporal delay value of 1 in Eq. 14 with δ and α with $\alpha\delta$ (we'll eventually take $\delta \rightarrow 0$). In addition differentiate Eq. 12 with respect to t to get

$$\begin{aligned}\frac{dq(x_i,t)}{dt} \;&=\; -q_i(x_i,t)[\frac{d\Phi_i(x_i,t)}{dt} - \\ &\qquad \int dx'_i q_i(x'_i,t)\frac{d\Phi_i(x'_i,t)}{dt}]. \qquad (15)\end{aligned}$$

In addition, in the $\delta \rightarrow 0$ limit, assuming q is a continuous function of t, Eq. 14 becomes

$$\frac{d\Phi_i(x_i,q)}{dt} \;=\; \alpha[\phi_i(x_i,q) - \Phi_i(x_i,q)]. \qquad (16)$$

where from now on the t variable is being suppressed for clarity.

If we knew the dynamics of ϕ_i, we could solve Eq. 16 via integrating factors, in the usual way. Instead, here we'll plug that equation for $\frac{d\Phi_i}{dt}$ into Eq. 15. Then use Eq. 12 to write $\Phi_i(x_i,q) =$ constant $-\ln(q_i(x_i))$. The result is

$$\begin{aligned}\frac{dq_i(x_i)}{dt} \;&=\; \alpha q_i(x_i)\; [\phi_i(x_i,q) + \ln(q_i(x_i))] \\ &\qquad -\alpha \int dx'_i\; q_i(x'_i)[\phi_i(x'_i) + \ln(q_i(x'_i))]. (17)\end{aligned}$$

C. Relation with Nearest Newton descent and replicator dynamics

As mentioned previously, there are many ways to find equilibria, and in particular many distributed algorithms for doing so. This is especially so in team games, where finding such equilibria reduces to descending a single over-arching Lagrangian.

One natural idea for descent in such games is to use the Newton-Raphson descent algorithm. However that algorithm cannot be applied directly to search across q in a distributed fashion, due to the need to invert matrices coupling the agents. As an alternative, one can consider what new distribution p the Newton algorithm would step to if there was no restriction that p be a product distribution. One can then ask what product distribution is closest to p, according to Kullback-Leibler distance [1]. It turns out that one *can* solve for that optimal product distribution. The associated update rule is called the **Nearest Newton** algorithm [31].

It turns out that when one writes down the Nearest Newton update rule, it says to replace each component $q_i(x_i)$ with the exact quantity appearing on the right-hand side of Eq. 17, where α is the stepsize of the update, and $\phi_i(x_i,t) = \beta E(G \mid x_i, q_{(i)}(t))$, as in parallel Brouwer updating for a team game [49]. In other words, in team games, the continuum limit of having each player using (bounded rational) best response is identical to the continuum limit of the Newton-Raphson algorithm for descending the Lagrangian, with the data-aging parameter α giving the stepsize.

Eq. 17 arises in other yet other contexts as well. In particular, say Φ_i is conditional expected rewards (i.e., $\phi_i(x_i, t-1) = E(g_i \mid q(., t-1))$). Then the $\beta \to \infty$ limit of Eq. 17 reduces to a simplified form of the replicator dynamics equation of evolutionary game theory [32, 33]. (If the stepsize α is an appropriately increasing function of $E(G)$ other versions of that dynamics arise.) This is because in that limit the ln term disappears, and the righthand side of Eq. 17 involves only the difference between player i's expected cost and the average expected cost of all players. This 3-way connection suggests using some of the techniques for solving replicator dynamics to expedite either parallel Brouwer or Nearest Newton.

D. Convergence and equilibria

By Eq. 17, at equilibrium, for each i, $q_i(x_i)[\phi_i(x_i, q) + \ln(q_i(x_i))]$ must be independent of i. One way this can occur is if it equals 0. However $q_i(x_i)$ can never be 0, by Eq. 12. This means we have an equilibrium at $q_i(x_i) \propto e^{-\phi_i(x_i,q)}$. Intuitively, this is exactly what we want, according to Eq. 12 and our interpretation of $\phi_i(x_i, q)$ as an estimate of $\phi_i(x_i, q)$. Note also that this solution means that $\phi_i(x_i, q) = \Phi_i(x_i, q)$, so that (according to Eq. 16) $\Phi_i(x_i, q)$ has also reached an equilibrium.

When our equilibrium has $q_i(x_i)[\phi_i(x_i, q) + \ln(q_i(x_i))] = A \neq 0$, we have

$$q_i(x_i) \;\propto\; e^{-q_i(x_i)\phi_i(x_i,q)}. \tag{18}$$

In light of Eq. 12, this means that $\Phi_i(x_i, q) \neq \phi_i(x_i, q)$. So by Eq. 16, $\Phi_i(x_i, q)$ hasn't reached an equilibrium in this case:

$$\frac{d\Phi_i(x_i, q)}{dt} \;=\; \alpha\phi_i(x_i, q)[1 - q_i(x_i)]. \tag{19}$$

If both $q_i(x_i)$ and $\phi_i(x_i, q)$ were frozen at this point, this solution for $\Phi_i(x_i, q)$ would not obey Eq. 14. So either $q_i(x_i)$ and/or $\phi_i(x_i, q)$ cannot be frozen. In fact, if $\phi_i(x_i, q)$ varies with time, then we know by Eq. 17 that $q_i(x_i)$ varies as well. So in either case $q_i(x_i)$ must vary, i.e., this equilibrium is not stable.

Although the dynamics has the desired fixed point, it may take a long time to converge there. There are several ways to analyze that: One is to examine the second derivatives (with respect to time) of the q_i and/or the Φ_i. Another is to examine the time-dependence of the residual error,

$$r_i^{ge}(x_i, t) \;\equiv\; \frac{e^{-\Phi_i(x_i,t)}}{\int dx'_i e^{-\Phi_i(x'_i,t)}} - \frac{e^{-\phi_i(x_i,t)}}{\int dx'_i e^{-\phi_i(x'_i,t)}}. \tag{20}$$

The next subsection includes a convergence analysis involving residual errors, but for a different variant of Brouwer from the ones considered so far.

E. Other variants of Brouwer updating

Note that data-aging can be viewed as moving only part-way from the current Φ_i to what it should be (i.e. to ϕ_i). As an alternative, one can dispense with the Φ_i and ϕ_i altogether, and instead step part-way from the current q to what it should be. This is partial movement to the (bounded rational) best response mixed strategy.

Formally, this means replacing Eq. 12 so that the update is not implicit, in how $\Phi_i(x_i, t)$ depends on the past value of $q(t-1)$ (Eq. 14), but explicit:

$$\begin{aligned} q_i(x_i, t) \;&=\; q_i(x_i, t-1) + \\ &\qquad \alpha[h_i(x_i, q_{(i)}(t-1)) - q_i(x_i, t-1)] \end{aligned} \tag{21}$$

where $h_i(x_i, q_{(i)}(t))$ is the Boltzmann distribution of what $q_i(x_i, t)$ would be, under ideal circumstances, and we implicitly have small stepsize α.

The only fixed point of this updating rule is where $q_i = h_i \;\forall i$. So just like with continuum-limit parallel Brouwer, we have the correct equilibrium. To investigate how fast the update rule of Eq. 21 arrives at that equilibrium, write its error at time t as the residual

$$\begin{aligned} r_i^{st}(x_i, t) \;&=\; q_i(x_i, t) - h_i(x_i, q_{(i)}(t)) \\ &=\; q_i(x_i, t-1)[1-\alpha] \;+\; \alpha h_i(x_i, q_{(i)}(t-1)) \\ &\qquad - \; h_i(x_i, q_{(i)}(t)) \\ &=\; q_i(x_i, t-1)[1-\alpha] \;+\; \alpha h_i(x_i, q_{(i)}(t-1)) \\ &\qquad - \; h_i[x_i, q_{(i)}(t-1) + \\ &\qquad\qquad \alpha[h_{(i)}(q(t-1)) - q_{(i)}(t-1)]] \end{aligned} \tag{22}$$

where we have assumed that all all players other than i are updating themselves in the same that i does (i.e., via Eq. 21), and $h_{(i)}(q(t-1))$ means the vector of the values of all $h_{j \neq i}(x_j)$ evaluated for $q(t-1)$.

With obvious notation, rewrite Eq. 22 as

$$\begin{aligned} r_i^{st}(x_i, t) \;&=\; q_i(x_i, t-1)[1-\alpha] \\ &\quad + \; \alpha h_i(x_i, q_{(i)}(t-1)) \\ &\quad - \; h_i[x_i, \;\; q_{(i)}(t-1) - \alpha r_{(i)}(t-1)] \end{aligned} \tag{23}$$

Now use the fact that α is small to expand the last h_i term on the righthand side to first order in its second (vector-valued) argument, getting the final result

$$r_i^{st}(x_i, t) \;\approx\; r_i(x_i, t)[1-\alpha] \;+\; \alpha \nabla h_i \cdot r_{(i)}(t-1) \tag{24}$$

where the gradient of h_i is with respect to the vector components of its second argument. Accordingly, if $r_i^{st}(x_i)$ starts much larger than the other residuals, it will be pushed down to their values. Conversely, if it starts much smaller than them, it will rise.

There are other ways one can reduce a stochastic game to a deterministic continuum-time process besides those considered here. In particular, this can be done in closed form for ficticious play games and some simple variants of it [16, 34].

V. STATISTICALLY COUPLING THE PLAYERS

A. The semicoordinate system of a game

Consider a multi-stage game like chess, with the stages (i.e., the instants at which one of the players makes a move) delineated by t. Now strategies are what are set by the players before play starts. So in such a multi-stage game the strategy of player i, x_i, must be the set of t-indexed maps taking what that player has observed in the stages $t' < t$ into its move at stage t. Formally, this set of maps is called player i's **normal form** strategy.

The joint strategy of the two players in chess sets their joint move-sequence, though in general the reverse need not be true. In addition, one can always find a joint strategy to result in any particular joint move-sequence. Now typically at any stage there is overlap in what the players have observed over the preceding stages. This means that even if the players' strategies are statistically independent, their move sequences are statistically coupled. In such a situation, by parameterizing the space Z of joint-move-sequences z with joint-strategies x, we shift our focus from the coupled distribution $P(z)$ to the decoupled product distribution, $q(x)$. This is the advantage of casting multi-stage games in terms of normal form strategies.

More generally, any onto mapping $\zeta : x \rightarrow z$, not necessarily invertible, is called a **semicoordinate system**. The identity mapping $z \rightarrow z$ is a trivial example of a semicoordinate system. Another example is the mapping from joint-strategies in a multi-stage game to joint move-sequences is an example of a semicoordinate system. In other words, changing the representation space of a multi-stage game from move-sequences z to strategies x is a semicoordinate transformation of that game.

We can perform a semicoordinate transformation even in a single-stage game. Say we restrict attention to distributions over X that are product distributions. Then changing $\zeta(.)$ from the identity map to some other function means that the players' moves are no longer independent. After the transformation their move choices — the components of z — are statistically coupled, even though we are considering a product distribution.

Formally, this is expressed via the standard rule for transforming probabilities,

$$P_Z(z \in Z) \equiv \zeta(P_X) \equiv \int dx P_X(x) \delta(z - \zeta(x)), \quad (25)$$

where P_X and P_Z are the distributions across X and Z, respectively. To see what this rule means geometrically, let $\mathcal{P}$ be the space of all distributions (product or otherwise) over Z. Recall that $\mathcal{Q}$ is the space of all product distributions over X, and let $\zeta(\mathcal{Q})$ be the image of $\mathcal{Q}$ in $\mathcal{P}$. Then by changing $\zeta(.)$, we change that image; different choices of $\zeta(.)$ will result in different manifolds $\zeta(\mathcal{Q})$.

As an example, say we have two players, with two possible moves each. So z consists of the possible joint moves, labeled $(1, 1), (1, 2), (2, 1)$ and $(2, 2)$. Have $X = Z$, and choose $\zeta(1, 1) = (1, 1)$, $\zeta(1, 2) = (2, 2)$, $\zeta(2, 1) = (2, 1)$, and $\zeta(2, 2) = (1, 2)$. Say that q is given by $q_1(x_1 = 1) = q_2(x_2 = 1) = 2/3$. Then the distribution over joint-moves z is $P_Z(1, 1) = P_X(1, 1) = 4/9$, $P_Z(2, 1) = P_Z(2, 2) = 2/9$, $P_Z(1, 2) = 1/9$. So $P_Z(z) \neq P_Z(z_1)P_Z(z_2)$; the moves of the players are statistically coupled, even though their strategies x_i are independent.

Such coupling of the players' moves can be viewed as a manifestation of sets of potential binding contracts. To illustrate this return to our two player example. Each possible value of a component x_i determines a pair of possible joint moves. For example, setting $x_1 = 1$ means the possible joint moves are $(1, 1)$ and $(2, 2)$. Accordingly such a value of x_i can be viewed as a set of proffered binding contracts. The value of the other components of x determines which contract is accepted; it is the intersection of the proffered contracts offered by all the components of x that determines what single contract is selected. Continuing with our example, given that $x_1 = 1$, whether the joint-move is $(1, 1)$ or $(2, 2)$ (the two options offered by x_1) is determined by the value of x_2.

B. Representational properties

Binding contracts are a central component of cooperative game theory. In this sense, semicoordinate transformations can be viewed as a way to convert noncooperative game theory into a form of cooperative game theory. Indeed, any cooperative mixed strategy can be cast as a non-cooperative game mixed strategy followed by an appropriate semicoordinate transformation. Formally, any P_Z, no matter what the coupling among its components, can be expressed as $\zeta(P_X)$ for some product distribution P_X for and associated $\zeta(.)$ [50]

Less trivially, given any model class of distributions $\{P_Z\}$, there is an X and associated $\zeta(.)$ such that $\{P_Z\}$ is identical to $\zeta(\mathcal{Q}_X)$. Formally this is expressed in a result concerning Bayes nets. For simplicity, restrict attention to finite Z. Order the components of Z from 1 to N. For each index $i \in \{1, 2, \ldots, N\}$, have the parent function $\mathcal{P}(i, z)$ fix a subset of the components of z with index greater than i, returning the value of those components for the z in its second argument if that subset of components is non-empty. So for example, with $N > 5$, we could have $\mathcal{P}(1, z) = (z_2, z_5)$. Another possibility is that $\mathcal{P}(1, z)$ is the empty set, independent of z.

Let $A(\mathcal{P})$ be the set of all probability distributions P_Z that obey the conditional dependencies implied by $\mathcal{P}$: $\forall\ P_Z \in A(\mathcal{P}), z \in Z$,

$$P_Z(z) \ = \ \prod_{i=1}^{N} P_Z(z_i \mid \mathcal{P}(i, z)). \quad (26)$$

(By definition, if $\mathcal{P}(i, z))$ is empty, $P_Z(z_i \mid \mathcal{P}(i, z))$ is just the i'th marginal of P_Z, $P_Z(z_i)$.) Note that any distribution P_Z is a member of $A(\mathcal{P})$ for some $\mathcal{P})$ — in the worst case, just choose the exhaustive parent function $\mathcal{P}(i,z) = \{z_j : j > i\}$.

For any choice of $\mathcal{P}$ there is an associated set of distributions $\zeta(\mathcal{Q}_X)$ that equals $A(\mathcal{P})$ exactly:

Theorem: Define the components of X using multiple indices: For all $i \in \{1, 2, \ldots, N\}$ and possible associated values (as one varies over $z \in Z$) of the vector $\mathcal{P}(i,z)$, there is a separate component of x, $x_{i;\mathcal{P}(i,z)}$. This component can take on any of the values that z_i can. Define $\zeta(.)$ recursively, starting at $i = N$ and working to lower i, by the following rule: $\forall\, i \in \{1, 2, \ldots, N\}$,

$$[\zeta(x)]_i \;=\; x_{i;\mathcal{P}(i,z)}.$$

Then $A(\mathcal{P}) = \zeta(\mathcal{Q}_X)$.

Proof: First note that by definition of parent functions, due to the fact that we're iteratively working down from higher i's to lower ones, $\zeta(x)$ is properly defined. Next plug that definition into Eq. 25. For any particular x and associated $z = \zeta(x)$, those components of x that do not "match" z by having their second index equal $\mathcal{P}(i,z)$ get integrated out. After this the integral reduces to

$$P_Z(z) \;=\; \prod_{i=1}^{N} P_X([x_{i;\mathcal{P}(i,z)}] = z_i),$$

i.e., is exactly of the form stipulated in Eq. 26. Accordingly, for any fixed x and associated $z = \zeta(x)$, ranging over the set of all values between 0 and 1 for each of the distributions $P_X([x_{i;\mathcal{P}(i,z)} = z_i)$ will result in ranging over all values for the distribution $P_Z(z)$ that are of the form stipulated in Eq. 26. This must be true for all x. Accordingly, $\zeta(\mathcal{Q}_X) \subseteq A(\mathcal{P})$. The proof that $A(\mathcal{P}) \subseteq \zeta(\mathcal{Q}_X)$ goes similarly: For any given P_Z and z, simply set $P_X([x_{i;\mathcal{P}(i,z)}] = z_i)$ for all the independent components $x_{i;\mathcal{P}(i,z)}$ of x and evaluate the integral in Eq. 25. **QED.**

Intuitively, each component of x in the lemma is the conditional distribution $P_Z(z_i \mid \mathcal{P}(i,z))$ for some particular instance of the vector $\mathcal{P}(i,z))$. The lemma means that in principle we never need consider coupled distributions. It suffices to restrict attention to product distributions, so long as we use an appropriate semicoordinate system. In particular, mixture models over Z can be represented this way.

C. Maxent Lagrangians over X rather than Z

While the distribution over X uniquely sets the distribution over Z, the reverse is not true. However so long as our Lagrangian directly concerns the distribution over X rather than the distribution over Z, by minimizing that Lagrangian we set a distribution over Z. In this way we can minimize a Lagrangian involving product distributions, even though the associated distribution in the ultimate space of interest is not a product distribution.

The Lagrangian we choose over X should depend on our prior information, as usual. If we want that Lagrangian to include an expected value over Z (e.g., of a cost function), we can directly incorporate that expectation value into the Lagrangian over X, since expected values in X and Z are identical: $\int dz P_Z(z) A(z) = \int dx P_X(x) A(\zeta(x))$ for any function $A(z)$. (Indeed, this is the standard justification of the rule for transforming probabilities, Eq. 25.)

However other functionals of probability distributions can differ between the two spaces. This is especially common when $\zeta(.)$ is not invertible, so X is larger than Z. In particular, while the expected cost term is the same in the X and Z maxent Lagrangians, this is not true of the two entropy terms in general; typically the entropy of a $q \in \mathcal{Q}$ will differ from that of its image, $\zeta(q) \in \zeta(\mathcal{Q})$ in such a case.

More concretely, the fully formal definition of entropy includes a prior probability μ: $S_X \equiv \int dx p(x) \ln(\frac{p(x)}{\mu(x)})$, and similarly for S_Z. So long as $\mu(x)$ and $\mu(z)$ are related by the normal laws for probability transformations, as are $p(x)$ and $p(z)$, then *if the cardinalities of X and Z are the same*, $S_Z = S_X$ [51]. When the cardinalities of the spaces differ though (e.g., when X and Z are both finite but with differing numbers of elements), this need no longer be the case. The following result bounds how much the entropies can differ in such a situation:

Theorem: For all $z \in Z$, take $\mu(x)$ to be uniform over all x such that $\zeta(x) = z$. Then for any distribution $p(x)$ and its image $p(z)$,

$$-\int dz\; p(z)\ln(K(z)) \;\leq\; S_X - S_Z \;\leq\; 0,$$

where $K(z) \equiv \int dx \delta(z - \zeta(x))$. (Note that for finite X and Z, $K(z) \geq 1$, and counts the number of x with the same image z.) If we ignore the μ terms in the definition of entropy, then instead we have

$$0 \;\leq\; S_X - S_Z \;\leq\; -\int dz\; p(z)\ln(K(z)).$$

Proof: Write

$$\begin{aligned}
S_X \;&=\; -\int dz \int dx\; \delta(z - \zeta(x))\; p(x)\; \ln[\frac{p(x)}{\mu(x)}] \\
&=\; -\int dz \int dx\; \delta(z - \zeta(x))\; p(x)\;\; \times \\
&\qquad\qquad (\ln[\frac{p(x)}{d(z)\mu(x)}] + \ln[d(z)]) \\
&=\; -\int dz\; p(z) \ln[d(z)] \;\; - \\
&\qquad \int dz \int dx\; \delta(z - \zeta(x))\; p(x)\; \ln[\frac{p(x)}{d(z)\mu(x)}]
\end{aligned}$$

where $d_z \equiv \int dx \; \delta(z - \zeta(x)) \; \frac{p(x)}{\mu(x)}$. Define μ^z to be the common value of all $\mu(x)$ such that $\zeta(x) = z$. So $\mu(z) = \mu^z K(z)$ and $p(z) = \mu^z d(z)$. Accordingly, expand our expression as

$$\begin{aligned} S_X \;&=\; -\int dz \; p(z) \ln[\frac{p(z)}{\mu(z)}] \;-\; \int dz \; p(z)K(z) \;- \\ &\qquad \int dz \int dx \; \delta(z - \zeta(x)) \; p(x) \ln[\frac{p(x)}{d(z)\mu(x)}] \\ &=\; S_Z - \int dz \; p(z)K(z) \;+ \\ &\qquad \int dz \; p(z) \; (-\int dx \; \delta(z - \zeta(x)) \; \frac{p(x)}{p(z)} \ln[\frac{p(x)}{p(z)}]). \end{aligned}$$

The x-integral of the right-hand side of the last equation is just the entropy of normalized the distribution $\frac{p(x)}{p(z)}$ defined over those x such that $\zeta(x) = z$. Its maximum and minimum are $\ln[K(z)]$ and 0, respectively. This proves the first claim. The second claim, where we "ignore the μ terms", is proven similarly. **QED.**

In such cases where the cardinalities of X and Z differ, we have to be careful about which space we use to formulate our Lagrangian. If we use the transformation $\zeta(.)$ as a tool to allow us to analyze bargaining games with binding contracts, then the direct space of interest is actually the x's (that is the place in which the players make their bargaining moves). In such cases it makes sense to apply all the analysis of the preceding sections exactly as it is written, concerning Lagrangians and distributions over x rather than z (so long as we redefine cost functions to implicitly pre-apply the mapping $\zeta(.)$ to their arguments). However if we instead use $\zeta(.)$ simply as a way of establishing statistical dependencies among the moves of the players, it may make sense to include the entropy correction factor in our x-space Lagrangian.

An important special case is where the following three conditions are met: Each point z is the image under $\zeta(.)$ of the same number of points in x-space, n; $\mu(x)$ is uniform (and therefore so is $\mu(z)$); and the Lagrangian in x-space, $\mathscr{L}_x$, is a sum of expected costs and the entropy. In this situation, consider a z-space Lagrangian, $\mathscr{L}_z$, whose functional dependence on P_z, the distribution over z's, is identical to the dependence of $\mathscr{L}_x$ on P_x, except that the entropy term is divided by n [52]. Now the minimizer $P^*(x)$ of $\mathscr{L}_x$ is a Boltzmann distribution in values of the cost function(s). Accordingly, for any z, $P^*(x)$ is uniform across all n points $x \in \zeta^{-1}(z)$ (all such x have the same cost value(s)). This in turn means that $S(\zeta(P_x)) = nS(P_z)$. So our two Lagrangians give the same solution, i.e., the "correction factor" for the entropy term is just multiplication by n.

D. Semicoordinate transformations in team games

Now consider situations in which one wishes to find the global minimum of the Lagrangian for a team game. To illustrate the generality of the arguments, situations where one has to to use Monte Carlo estimates of conditional expectation values to descend the shared Lagrangian (rather than evaluate them closed-form) will be considered.

Say we are currently at a local minimum $q \in \mathcal{Q}$ of $\mathscr{L}$ of the team game. Usually we can break out of that minimum by raising β and then resuming the updating; typically changing β changes $\mathscr{L}$ so that the Lagrange gaps are nonzero. So if we want to anneal β anyway (e.g., to find a minimum of the shared cost function G), it makes sense to do so to break out of any local minima.

There are many other ways to break out of local minima without changing the Lagrangian (as we would if we changed β, for example) [31]. Here we show how to use semicoordinate transformations to do this. As explicated below, they also provide a general way to lower the value of the Lagrangian, whether or not one has local minimum problems.

Say our original semicoordinate system is $\zeta^1(.)$. Switch to a different semicoordinate system $\zeta^2(.)$ for Z and consider product distributions over the associated space X^2. Geometrically, the semicoordinate transformation means we change to a new submanifold $\zeta^2(\mathcal{Q}) \subset \mathcal{P}$ without changing the underlying mapping from $p(z)$ to $\mathscr{L}_Z(p)$.

As a simple example, say ζ^2 is identical to ζ^1 except that it joins two components of x into an aggregate semicoordinate. Since after that change we can have statistical dependencies between those two components, the product distributions over X^2, $\zeta^2(\mathcal{Q}_{X^2})$, map to a superset of $\zeta^1(\mathcal{Q}_{X^1})$. Typically the local minima of that superset do not coincide with local minima of $\zeta^1(\mathcal{Q}_{X^1})$. So this change to X^2 will indeed break out of the local minimum, in general.

More care is needed when working with more complicated semicoordinate transformations. Say before the transformation we are at a point $p^* \in \zeta^1(\mathcal{Q}_{X^1})$. Then in general p^* will not be in the new manifold $\zeta^2(\mathcal{Q}_{X^2})$, i.e., p^* will not correspond to a product distribution in our new semicoordinate system. (This reflects the fact that semicoordinate transformations couple the players.) Accordingly, we must change from p^* to a new distribution when we change the semicoordinate system.

To illustrate this, say that the semicoordinate transformation is bijective. Formally, this means that $X^2 = X^1 \equiv X$ and $\zeta^2(x) = \zeta^1(\xi(x))$ for a bijective $\xi(.)$. Have $\xi(.)$, the mapping from X^2 to X^1, be the identity map for all but a few of the M total components of X, indicated as indices $1 \rightarrow n$. Intuitively, for any fixed $x^2_{n+1 \rightarrow M} = x_{n+1 \rightarrow M}$, the effect of the semicoordinate transformation to $\zeta^2(.)$ from $\zeta^1(.)$ is merely to "shuffle" the associated mapping taking semicoordinates $1 \rightarrow n$ to Z, as specified by $\xi(.)$. Moreover, since $\xi(.)$ is a bijection, the maxent Lagrangians over X^1 and X^2 are identical: $\mathscr{L}_{X^1}(\xi(p^{X^2})) = \mathscr{L}_{X^2}((p^{X^2}))$.

Now say we set $q^{X^2}_{n+1 \rightarrow M} = q^{X}_{n+1 \rightarrow M}$. This means we can estimate the expectations of G conditioned on possible $x^2_{1 \rightarrow n}$ from the Monte Carlo samples conditioned on $\xi(x^2_{1\to n})$. In particular, for any $\xi(.)$ we can estimate $E(G)$ as $\int dx^2_{1\to n} p^{X^2}(x^2_{1\to n}) E(G \mid \xi(x^2_{1,\ldots,n}))$ in the usual way. Now entropy is the sum of the entropy of semicoordinates $n+1 \to M$ plus that of semicoordinates $1 \to n$. So for any choice of $\xi(.)$ and $q^{X^2}_{1\to n}$, we can approximate $\mathscr{L}_X = \mathscr{L}_{X^2}$ as (our associated estimate of) $E(G)$ minus the entropy of $p^{X^2}_{1\to n}$, minus a constant unaffected by choice of $\xi(.)$.

So for finite and small enough cardinality of the subspace $|X_{1\to n}|$, we can use our estimates $E(G \mid \xi(x^2_{1\to n}))$ to search for the "shuffling" $\xi(.)$ and distribution $q^{X^2}_{1\to n}$ that minimizes $\mathscr{L}^X$ [53]. In particular, say we have descended $\mathscr{L}_X$ to a distribution $q^{X^1}(x) = q^*(x)$. Then we can set $q^{X^2} = q^*$, and consider a set of of "shuffling $\xi(.)$". Each such $\xi(.)$ will result in a different distribution $q^{X^1}(x) = q^{X^2}(\xi^{-1}(x)) = q^*(\xi^{-1}(x))$. While those distributions will have the same entropy, typically they will have different (estimates of) $E(G)$ and accordingly different local minima of the Lagrangian.

Accordingly, searching across the $\xi(.)$ can be used to break out of a local minimum. However since $E(G)$ changes under such transformations even if we are not at a local minimum we can searching across $\xi(.)$, as a new way (in addition to those discussed above) for lowering the value of the Lagrangian. Indeed, there is always a bijective semicoordinate transformation that reduces the Lagrangian: simply choose $\xi(.)$ to rearrange the $G(x)$ so that $G(x) < G(x') \Leftrightarrow q(x) < q(x')$. In addition one can search for that $\xi(.)$ in a distributed fashion, where one after the other each agent i rearranges its semicoordinate to shrink $E(G)$. Furthermore to search over semicoordinate systems we don't need to take any additional samples of G. (The existing samples can be used to estimate the $E(G)$ for each new system.) So the search can be done off-line.

To determine the semicoordinate transformation we can consider other factors besides the change in the value of the Lagrangian that immediately arises under the transformation. We can also estimate the the amount that subsequent evolution under the new semicoordinate system will decrease the Lagrangian. We can estimate that subsequent drop in a number of ways: the sum of the Lagrangian gaps of all the agents, gradient of the Lagrangian in the new semicoordinate system, etc.

E. Distributions over semicoordinate systems

The straightforward way to implement these kinds of schemes for finding a good semicoordinate systems is via exhaustive search, hill-climbing, simulated annealing, or the like. Potentially it would be very useful to instead find a new semicoordinate system using search techniques designed for continuous spaces. When there are a finite number of semicoordinate systems (i.e., finite X and Z) this would amount to using search techniques for continuous space to optimize a function of a variable having a finite number of values. However we now know how to do that: use PD theory. In the current context, this means placing a product probability distribution over a set of variables parameterizing the semicoordinate system, and then evolving the probability distribution.

More concretely, write

$$\begin{aligned}\mathscr{L}(q) &= \beta \sum_\theta \sum_x P(\theta) \prod_{i=1}^N q_i(x_i) G(\zeta(x,\theta)) + S(q) \qquad (27)\\ &= \beta \sum_\theta \sum_x \prod_{i=1}^N q_i(x_i) P(\theta) G(\zeta(x,\theta)) + S(q)\end{aligned}$$

where θ is a parameter on the semicoordinate system. We can rewrite this using an additional semicoordinate transformation, as

$$\mathscr{L}(q^*) = \beta \sum_{x^*} \prod_{i=1}^{N+1} q^*_i(x^*_i) G(\zeta(x^*)) + S\ (q^*) \qquad (28)$$

where $x^*_i = x_i$ for all i up to N, and $x^*_{N+1} = \theta$. (As usual, depending on what space we cast our Lagrangian in, the entropy can either have the argument of the entropy term starred — as here — or not.)

Intuitively, this approach amounts to introducing a new coordinate/agent, whose "job" is to set the semicoordinate system governing the mapping from the other agents to a z value. This provides an alternative to periodically (e.g., at a local minimum) picking a set of alternative semicoordinate systems and estimating which gives the biggest drop in the overall Lagrangian. We can instead use Nearest Newton, Brouwer updating, or what have you, to continuously search for the optimal coordinate system as we also search for the optimal x. The tradeoff, of course, is that by introducing an extra coordinate/agent, we raise the noise level all the original semicoordinates experience. (This raises the issue of what best parameterization of $\zeta(.)$ to use, an issue not addressed here.)

VI. RELATED WORK AND EXTENSIONS

The core of this paper is the maxent Lagrangian and associated Boltzmann distribution solution. These have been investigated for well over a century in the statistical physics. The use of the Boltzmann distribution over possible moves also has a long history in the RL literature. In all of this RL work though the Boltzmann distribution is usually motivated either as an *a priori* reasonable way to trade off exploration and exploitation, as part of Markov Chain Monte Carlo procedure, or by its asymptotic convergence properties [30].

Independent of the work in [4], the maxent Lagrangian and/or the Boltzmann distribution has previously been suggested as a way to model human players [16, 34, 35]. Some of that work has explicitly noted the relation between the Boltzmann distribution and statistical physics

[36]. However the motivation of the maxent Lagrangian and Boltzmann distribution in that work is *ad hoc*, based on particular simple models of human decision-making and/or of player interactions. There is no use of information theory to derive the maxent Lagrangian from first principles, as is done in PD theory.

Some of the benefits of such a first principles approach are presented in this paper. Others are reported in [4]. These include an explicit term in the analysis that, in light of information theory, corresponds to cost of computation. Other benefits are natural ways to accommodate multiple cost functions per player. PD theory also highlights the very close relationship betweeen bounded rational game theory and statistical physics. This relationship allows many of the tools of statistical physics to be applied to bounded rational games. For example, by exploiting the grand canonical ensemble of stastistical physics, they allow one to analyze bounded rational games with variable numbers of players — in essence, a bounded rational extension of evolutionary game theory [4].

Finally, it's important to note that PD theory has many applications beyond those considered in this paper. For example, see [8, 31, 37–40] for other work relating the maxent Lagrangian to distributed control and to distributed optimization. See [31] for algorithms for speeding up convergence to bounded rational equilibria. Some of those algorithms are related to simulated and deterministic annealing [41]. See also [42, 43] for work showing, respectively, how to use PD theory to improve Metropolis-Hastings sampling and how to extend it to continuous move spaces and time-extended strategies.

Acknowledgements: I would like to thank Stefan Bieniawski, Bill Macready, Stephane Airiau, Chiu Fan Lee, George Judge, Chris Henze, and Ilan Kroo for helpful discussion.

[1] T. Cover and J. Thomas, *Elements of Information Theory* (Wiley-Interscience, New York, 1991).

[2] D. Mackay, *Information theory, inference, and learning algorithms* (Cambridge University Press, 2003).

[3] E. T. Jaynes and G. L. Bretthorst, *Probability Theory : The Logic of Science* (Cambridge University Press, 2003).

[4] D. H. Wolpert, in *Complex Engineering Systems*, edited by A. M. D. Braha and Y. Bar-Yam (2004).

[5] D. H. Wolpert (2003), cond-mat/0307630.

[6] D. H. Wolpert, *Bounded rationality game theory and information theory* (2004), submitted.

[7] W. Macready, S. Bieniawski, and D. Wolpert, *Adaptive multi-agent systems for constrained optimization* (2004), technical report IC-04-123.

[8] C. F. Lee and D. H. Wolpert, in *Proceedings of AAMAS 04* (2004).

[9] S. Bieniawski and D. H. Wolpert, in *Proceedings of AAMAS04* (2004).

[10] S. Bieniawski, D. H. Wolpert, and I. Kroo, in *Proceedings of 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York* (2004), in press.

[11] W. B. Arthur, The American Economic Review **84**(2), 406 (1994).

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 1998).

[13] L. P. Kaelbing, M. L. Littman, and A. W. Moore, Journal of Artificial Intelligence Research **4**, 237 (1996).

[14] R. H. Crites and A. G. Barto, in *Advances in Neural Information Processing Systems - 8*, edited by D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (MIT Press, 1996), pp. 1017–1023.

[15] J. Hu and M. P. Wellman, in *Proceedings of the Fifteenth International Conference on Machine Learning* (1998), pp. 242–250.

[16] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games* (MIT Press, Cambridge, MA, 1998).

[17] T. Basar and G. Olsder, *Dynamic Noncooperative Game Theory* (Siam, Philadelphia, PA, 1999), second Edition.

[18] M. Osborne and A. Rubenstein, *A Course in Game Theory* (MIT Press, Cambridge, MA, 1994).

[19] R. Aumann and S. Hart, *Handbook of Game Theory with Economic Applications* (North-Holland Press, 1992).

[20] D. Fudenberg and J. Tirole, *Game Theory* (MIT Press, Cambridge, MA, 1991).

[21] R. Axelrod, *The Evolution of Cooperation* (Basic Books, NY, 1984).

[22] T. Sandholm and V. R. Lesser, Artificial Intelligence **94**, 99 (1997).

[23] A. Neyman, Economics Letters **19**, 227 (1985).

[24] C. Boutilier, Y. Shoham, and M. P. Wellman, Artificial Intelligence Journal **94**, 1 (1997).

[25] N. I. Al-Najjar and R. Smorodinsky, Game and Economic Behavior **37**(26-39) (2001).

[26] A. Tversky and D. Kahneman, Journal of Risk and Uncertainty **5**, 297 (1992).

[27] D. Kahneman, American Economic Review (Proceedings) **93:2**, 162 (2003).

[28] G. Judge, D. Miller, and W. Cho, in *Ecological Inference:New methodological Strategies*, edited by King, Rosen, and Tanner (Cambridge University Press, 2004).

[29] E. T. Jaynes, Physical Review **106**, 620 (1957).

[30] C. Watkins and P. Dayan, Machine Learning **8**(3/4), 279 (1992).

[31] D. H. Wolpert and S. Bieniawski, in *Proceedings of CDC04* (2004).

[32] K. Tuyls, D. Heytens, A. Nowe, and B. Manderick, in *Lecture Notes in Artificial Intelligence, LNAI, (ECML 2003)* (2003).

[33] K. Verbeeck, A. Nowe, and K. Tuyls, in *Proceedings of AAMAS-3. University of Wales, Aberystwyth* (2003).

[34] J. Shamma and G. Arslan, *Dynamic fictitious play, dynamic gradient play, and distributed convergence to nash equilibria* (2004), submitted.

[35] D. Fudenberg and D. Kreps, Game and Economic Behavior **5**, 320 (1993).

[36] S. Durlauf, Proc. Natl. Acad. Sci. USA **96**, 10582 (1999).

[37] W. Macready and D. H. Wolpert, in *Proceedings of ICCS*

04 (2004).

[38] S. Airiau and D. H. Wolpert (2004), submitted to AAMAS 04.

[39] N. Antoine, S. Bieniawski, I. Kroo, and D. H. Wolpert, in *Proceedings of 42nd Aerospace Sciences Meeting* (2004), aIAA-2004-0622.

[40] S. Bieniawski and D. H. Wolpert, in *Proceedings of ICCS 04* (2004).

[41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd ed.)* (Wiley and Sons, 2000).

[42] D. H. Wolpert and C. F. Lee, *Adaptive metropolis hastings sampling using product distributions* (2004), submitted to ICCS04.

[43] D. H. Wolpert, in *Proceedings of MSRAS04*, edited by A. S. et al (Springer Verlag, 2004).

[44] Throughout this paper, the integral sign is implicitly interpreted as appropriate, e.g., as Lebesgue integrals, point-sums, etc.

[45] The subscript $q_{(i)}$ on the expectation value indicates that it is evaluated according the distribution $\prod_{j \neq i} q_j$.

[46] Parallel Brouwer updating can be done with minimal memory requirements on the agents. Say a particular move has just been taken by agent i, and that the most recent time it was taken before that was T iterations ago. Furthermore, say the cost recorded by i for that most recent instance by was r. Then the new estimated cost for that move, E', is related to the previous one, E, by $E' = \frac{r+k^T E a}{1+k^T a}$, where k is a constant less than 1, and a is initially set to 1, while itself also being updated according to $a+ = k^T$. So agent i only needs to keep a running tally of E, a, and T for each of its possible moves to use data-aging, rather than a tally of all historical time-cost pairs

[47] Proof outline: Write the entropy after the update as a sum of non-i entropies (which are unchanged by the update) plus i's new entropy. Then expand i's new entropy. This gives the value of the new Lagrangian as -ln$[N_i]$. Then do the subtraction.

[48] To write this as exponential data-aging set the exponent γ of such data-aging to -ln$(1 - \alpha)$).

[49] More generally, the Nearest Newton technique uses this update rule with $\phi_i(x_i, t) = \beta E(g_i \mid x_i, q_{(i)}(t))$ where each $g_i(x) = G(x) - D(x_{(i)})$ for some function D. See [31].

[50] In the worst case, one can simply choose X to have a single component, with $\zeta(.)$ a bijection between that component and the vector z — trivially, any distribution over such an X is a product distribution.

[51] For example, if $X = Z = \mathbb{R}$, then $\ln[\frac{p(\zeta(x))}{\mu(\zeta(x))}] = \ln[\frac{p(x)J_\zeta(x)}{\mu(x)J_\zeta(x)}] = \ln[\frac{p(x)}{\mu(x)}]$, where $J_\zeta(x)$ is the determinant of the Jacobian of $\zeta(.)$ evaluated at x. Accordingly, as far as transforming from X to Z is concerned, entropy is just a conventional expectation value, and therefore has the same value whichever of the two spaces it is evaluated in.

[52] For example, if $\mathscr{L}_x(P_x) = \beta E_{P_x}(G(\zeta(.))) - S(P_x)$, then $\mathscr{L}_z(P_z) = \beta E_{P_z}(G(.)) - S(P_z)/n$, where P_x and P_z are related as in Eq. 25.

[53] penalizing by the bias^2 plus variance expression if we intend to do more Monte Carlo — see [5].